## Sektion 3

Fråga 1

You have been invited as a software architect to a new project, which already has a development team with senior developers and a project manager. On the first introduction meeting with the team, you've been asked what is going to be your role as software architect, why is it important for the team and how are you going to collaborate with all team members, including senior developers and project manager manager. How would you answer to the team?

**What will be my role as a software architect:**

My role will be to design the software architecture for the given system. It is about choosing the right technology and tools, the right design patterns, and the right methodologies and processes for myself (as the architect) as well as for the team, since the software architecting process is usually a team effort. I will also report to project managers and communicate with stakeholders to achieve my goals.

**Why it is important for the team to have a software architect:**

It's important to align all team members on what to create. To have a shared vision of what the system looks like today, and how it will look like in the future, by the means of views and documentation. It's also important to create the right thing, and not just start developing straight away, which might end up with the wrong product being made. The architect makes sure the team makes what the stakeholders needs.

**How I am going to collaborate with all team members, including senior developers and project managers:**

I am going to collaborate with all/some team members in forms of certain activities, such as the game "DecidArch", drawing stakeholder maps, quality attribute maps, QA web, perform the architecture trade-off analysis method (ATAM), among other activities. I will also let the business analyst help the team by making business entity models and business process models, which will help me and the architecting team to do our job better. One way of communicating with the team (including project managers, sen. devs, etc) will be in the form of diagrams/views, such as module views, component&connector views, stakeholder views, and allocator views, etc. Certain views are more relevant to some people than others, for example stakeholder view is more interesting for the project manager than say the module view which is a bit more source code oriented.

## Sektion 4

**Fråga 1**

As a software architect, you are discussing product owner and project manager that you need to arrange a series of activities with stakeholders. However, both of them do not seem to understand why it is important, since you already have a very clear task to develop an application. Please, explain to product owner and project manager why do you need to engage with the stakeholders, what outcomes/information do you expect to receive, how are you going to identify stakeholders, and what exactly activities you are going to perform.

**My explanation to why we need to engage with stakeholders:**

We need to engage with the stakeholders in order to find their needs and (functional) requirements. Software needs to have a purpose and solve real world problems, and those problems, we can only understand if we understand our stakeholders.

**What outcomes/information do I expect to receive from this:**

I expect to receive functional requirements and use case stories/scenarios, from different use groups of our system, for example from regular users and administrators of the system.

**How am I going to identify stakeholders:**

I will, together with the team, draw a stakeholder map along with a stakeholder power-influence graph and stakeholder core/external diagram, to know what stakeholders are relevant for our system, and how much they should have a say in our system (power/influece or if they are a core part of the system, or just some side characters...)

**What activities I will perform:**

To find out the requirements of the stakeholders, I will simply ask them in stakeholder interviews.

## Sektion 5

Fråga 1

You have been tasked to produce relevant documentation for the software architecture you are developing, with the goal to inform company leadership on the proposed architecture and how it support business values and also to allow the development team to start the implementation work. How would you approach this problem? How will you select relevant views and what views are you going to make? Will you ask your development team to assist you in this task and if yes, how?

**How will I approach the problem of making documentation and how will I select relevant views:**

When an architectural proposal or decision is made, this will be written in a document called ADR (architecture decision record).

I will look into, and run the V&B process (views and beyond) to find the needed views.

**What views will I make:**

I will make stakeholder view, module view, component and connector view and allocator view.

**Will my dev team help me?**

Yes, I will ask some developers to help me in the making of different views, but specifically the module view, since this is what I believe to be most relevant for engineers as it is close to the actual source code. I will ask them to make UML class diagrams.

## Sektion 6

**Fråga 1**

After talking to all relevant stakeholders, company leadership, investors, and sales department, you end up with a long list of requests that the system you are developing has to be the best on the market, easiest to use for the users and work well with other systems, handle all requests in real time and be always online, has to allow implementing new features easily. How are you going to deal with this mess? How will you formulate all that for your development team and set priorities in the way that you actually address those requests that make sense?

**How I will formulate that we need to set priorities that address the requests that make sense:**

It's clear there's been a "shopping cart mentality" here, where everyone wants the absolute best, which is not realistic. We need to prioritise.

**How will I deal with this mess?**

I will ask my team to help me draw a Quality Attribute Web, and put the use case scenarios on the web, and perform voting to see what use cases we need to prioritise. I will ask them to follow one of two priorisation techniques:

MoSCoW: Priorisation according to: must have, should have, could have, won't have.

The Kano model: what are the baseline requirements (where customer is really dissatisfied without), what are the one-dimensional requirements (where customer is really dissatisfied without, but if implemented well, is greatly satisfied) and what are the attractive requirements (just having them makes the customer greatly satisfied).

From this process, we will end up with a quality attribute priorisation list, and a list of the most important use case scenarios.

Fråga 2

In the discussions on the overall architecture for your system with the development team, you have been asked to describe a publisher-subscriber architectural pattern and if it can provide certain benefits or drawbacks for the system in question. Please, provide a developer-friendly description of the pattern along with pros/cons and possible alternatives.

**Publish/subscribe description:**

The idea of publish/subscribe is similar to how YouTube works, a user (viewer, in this case "subscriber") subscribes to a channel (in publish/subscribe terms it's called a "topic"), and a YouTube content creator (in publish/subscribe terms, its the publisher) posts videos on the channel. When a post is made, the subscriber receives the conten in form of a notification.

Similarly, in publish/subscribe, a subscriber S makes a subscription to the topic X, and some publisher P posts data to the topic X, this will make it so that A will receive a callback with the data posted by P. S will only receive the data if they are subscribed to the topic X. The data is posted to an intermediary broker server, which keeps track of who is subscribed to what. Several publishers can post to the same topic. Usually, the data is not stored on the broker, but sent straight away to subscribers. Implementation example: MQTT (last message on a topic in stored in MQTT on the broker).

**Pros:**

Easy to implement and has great flexibility and modularity. It's also easy to understand since it's something we're already used to as a concept.

**Cons:**

Since publish/subscribe relies on a broker server, it will have a slight delay, and thus have a little lower performance than some alternatives. It also has a single point of failure, if the broker goes down, it will be big trouble because no messages will get delivered.

**Alternatives to publish/subscribe:**

A blackboard pattern, where data is pushed by anyone, and can be read by anyone. The data also stays there. (Similar to real life blackboards). This pattern requires some kind of polling, and thus probably lower performance than publish/subscribe which is more "event based" (deals with callbacks).

You could also opt for something like peer to peer patterns, client-server, etc. I don't have context for what the system in question is so I can't really give good recommendations here.

## Sektion 7

Fråga 1

After you have collected and sorted all possible information relevant for your project, you need to start actually creating architecture. You arranged a meeting with your development team to get started with this task. Please, give an introduction to all team members on what the overall architecting process is going to look like, what are the steps in the process, inputs and outputs, if the process is going to be phased or iterative. You can use the ADD process that was used during this course or offer an alternative. In any case, please give a sufficiently detailed description of the process and its steps.

*We will use the ADD process.*

**The inputs to get started are:** functional requirements, constraints, concerns, primary design decisions, quality attribute priorisation.

**The output will basically be:** software architecture. It will be in the form of a bunch of ADRs (architecture decision records).

**The steps in the process is broadly like this:**

You make sure all inputs that are necessary is in place. Then, you select a software architect element to decompose (for example something broad like "backend"). You take a look at different architectural drivers for the element and decide on what is best according to your input. You will then get a bunch of more software architecture elements out of this, for example "backend" is now decomposed into "client-server with a SQL-like database, communicating with the front-end using websockets,written in a high-performance systems language because xyz..." along with alternatives. (**Will the process to phased or iterative?**) Then, you re-iterate on this recursively, decomposing further, for example the "websockets communication" can be decomposed into "we will use the Socket.IO library and xyz ...". After enough details (we will not go too far into the details, we have to stop somewhere!), we will go back to the more broad software architecture elements, and start to decompose those (for example "frontend"). For each of those decisions that we make in the process, we also make an ADR, which is the main output of the process.

Fråga 2

After you created a sufficient architectural proposal for your system, the product owners asked you to present that the developed architecture actually addresses business needs and values. There is also a question on why the architecture is not fully complete and some technical details and views are missing. Please, provide a product owner with a non-technical description of the evaluation process, it's steps, and what benefits and proofs it delivers, as well as a rationale on the amount of architecting work that needs to be done before the development starts.

**Does the architecture actually address business needs and values?**

Yes, of course. We started by looking into those exact questions when we held the stakeholder interviews, to find what stakeholder values in our system, and what business needs they have. We then performed a set of processes and transformed their needs/values into software architecture.

**Why is the architecture not 100% complete, some views and tech details are missing?**

This is intentional, as we can not know everything up front, that would take alot of time and money. Also, some thing are yet subject to change, some stakeholders might slightly change their requirements/values, but things will appear to be "obvious" once we get there, and can show our product to the stakeholders.

**The evaluation process:**

The evaluation process we will use is the ATAM process (architecture trade-off analysis method). The process goes like this:

We invite relevant stakeholders to a meeting. We take each use case scenario, and discuss its difficulty to implement (according to the dev team) and its importance (according to the relevant stakeholder). We rank each scenario in difficulty (easy, medium, hard) and importance (low, medium, high). This process will find any items that are low importance and high difficulty, and those should be prioritised away, in favour of higher importance and easier difficulty items. The process will also proof how good or bad our architecture really is. If needed, the ATAM process will indicate to us if we need to go back to the drawing board on the architecture, or even on the business idea in the worse case.

**Amount of architecting work that needs to be done before development starts:**

This is a trade-off between architecture time VS development time. The more architecture investment, the less development time, and vice versa. The point here is to find the sweet spot. Don't go into too much details in the architecture. And don't start developing straight away without architecture, as you might have to re-develop certain things later as it appears that this is not what stakeholders wanted. Really, this comes down to experience, but a good rule of thumb could be to have the architecture 80% finished of the final product's architecture (which can be hard to know where 80% is...).